

Internet-Draft
Network Working Group
Intended Status: Informational
Expires: April 18, 2016

Bhuvaneswaran Vengainathan
Anton Basil
Veryx Technologies
Mark Tassinari
Hewlett-Packard
Vishwas Manral
Ionos Corp
Sarah Banks
VSS Monitoring
October 19, 2015

Terminology for Benchmarking SDN Controller Performance
draft-ietf-bmwg-sdn-controller-benchmark-term-00

Abstract

This document defines terminology for benchmarking an SDN Controller's performance. The terms provided in this document help to benchmark SDN controller's performance independent of the controller's supported protocols and/or network services. A mechanism for benchmarking the performance of SDN controllers is defined in the companion methodology document. These two documents provide a standard mechanism to measure and evaluate the performance of various controller implementations.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 18, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Term Definitions	4
2.1. SDN Terms	4
2.1.1. SDN Node	4
2.1.2. SDN Application	4
2.1.3. Flow	4
2.1.4. Northbound Interface	5
2.1.5. Southbound Interface	5
2.1.6. Controller Forwarding Table	5
2.1.7. Proactive Flow Provisioning Mode	6
2.1.8. Reactive Flow Provisioning Mode	6
2.1.9. Path	7
2.1.10. Standalone Mode	7
2.1.11. Cluster/Redundancy Mode	7
2.1.12. Asynchronous Message	8
2.1.13. Test Traffic Generator	8
2.2. Test Configuration/Setup Terms	9
2.2.1. Number of SDN Nodes	9
2.2.2. Test Iterations	9
2.2.3. Test Duration	9
2.2.4. Number of Cluster nodes	10
2.3. Benchmarking Terms	10
2.3.1. Performance	10
2.3.1.1. Network Topology Discovery Time	10
2.3.1.2. Asynchronous Message Processing Time.....	11
2.3.1.3. Asynchronous Message Processing Rate.....	11
2.3.1.4. Reactive Path Provisioning Time	12
2.3.1.5. Proactive Path Provisioning Time	12

2.3.1.6. Reactive Path Provisioning Rate	13
2.3.1.7. Proactive Path Provisioning Rate	13
2.3.1.8. Network Topology Change Detection Time	13
2.3.2. Scalability	14
2.3.2.1. Control Sessions Capacity	14
2.3.2.2. Network Discovery Size	14
2.3.2.3. Forwarding Table Capacity	15
2.3.3. Security	15
2.3.3.1. Exception Handling	15
2.3.3.2. Denial of Service Handling	16
2.3.4. Reliability	16
2.3.4.1. Controller Failover Time	16
2.3.4.2. Network Re-Provisioning Time	17
3. Test Coverage	17
4. References	18
4.1. Normative References	18
4.2. Informative References	19
5. IANA Considerations	19
6. Security Considerations	19
7. Acknowledgements	19
8. Authors' Addresses	19

1. Introduction

Software Defined Networking (SDN) is a networking architecture in which network control is decoupled from the underlying forwarding function and is placed in a centralized location called the SDN controller. The SDN controller abstracts the underlying network and offers a global view of the overall network to applications and business logic. Thus, an SDN controller provides the flexibility to program, control, and manage network behaviour dynamically through standard interfaces. Since the network controls are logically centralized, the need to benchmark the SDN controller performance becomes significant. This document defines terms to benchmark various controller designs for performance, scalability, reliability and security, independent of northbound and southbound protocols.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Term Definitions

2.1. SDN Terms

2.1.1. SDN Node

Definition:

An SDN node is a simulated/emulated entity that forwards data in a software defined environment.

Discussion:

An SDN node can be an emulated/simulated switch, router, gateway, or any network service appliance that supports standardized or proprietary programmable interface.

Measurement Units:

N/A

See Also:

None

2.1.2. SDN Application

Definition:

Any business logic that alter the network behaviour dynamically through controller's northbound interface.

Discussion:

SDN application can be any business application, cloud orchestration system, network services orchestration etc.,

Measurement Units:

N/A

See Also:

None

2.1.3. Flow

Definition:

A flow is a uni-directional sequence of packets having common properties derived from the data contained in the packet.

Discussion:

A flow can be set of packets having same source address, destination address, source port and destination port, or any of these combinations.

Measurement Units:
N/A

See Also:
None

2.1.4. Northbound Interface

Definition:

The northbound interface is the application programming interface provided by the SDN controller for the SDN services and applications to interact with the SDN controller.

Discussion:

The northbound interface allows SDN applications and orchestration systems to program and retrieve the network information through the SDN controller.

Measurement Units:
N/A

See Also:
None

2.1.5. Southbound Interface

Definition:

The southbound interface is the application programming interface provided by the SDN controller to interact with the SDN nodes.

Discussion:

Southbound interface enables controller to interact with the SDN nodes in the infrastructure for dynamically defining the traffic forwarding behaviour.

Measurement Units:
N/A

See Also:
None

2.1.6. Controller Forwarding Table

Definition:

A controller forwarding table contains flow entries learned in one of two ways: first, entries could be learned from traffic received through the data plane, or, second, these entries could be

statically provisioned on the controller, and distributed to devices via the southbound interface.

Discussion:

The controller forwarding table has an aging mechanism which will be applied only for dynamically learnt entries.

Measurement Units:

N/A

See Also:

None

2.1.7. Proactive Flow Provisioning Mode

Definition:

Controller programming flows in SDN nodes based on the flow entries provisioned through controller's northbound interface.

Discussion:

Orchestration systems and SDN applications can define the network forwarding behaviour by programming the controller using proactive flow provisioning. The controller can then program the SDN nodes with the pre-provisioned entries.

Measurement Units:

N/A

See Also:

None

2.1.8. Reactive Flow Provisioning Mode

Definition:

Controller programming flows in SDN nodes based on the traffic received from SDN nodes through controller's southbound interface

Discussion:

The SDN controller dynamically decides the forwarding behaviour based on the incoming traffic from the SDN nodes. The controller then programs the SDN nodes using Reactive Flow Provisioning.

Measurement Units:

N/A

See Also:

None

2.1.9. Path

Definition:

A path is a sequence of SDN nodes and links traversed by a flow.

Discussion:

As defined in RFC 2330, path is a sequence of the form $\langle h_0, l_1, h_1, \dots, l_n, h_n \rangle$, where $n \geq 0$, h_0 and h_n is a Host, $h_1 \dots h_{n-1}$ is an SDN Node, each l_i is a link between h_{i-1} and h_i . A pair $\langle l_i, h_i \rangle$ is termed a 'hop'. Note that path is a unidirectional concept.

Measurement Units:

N/A

See Also:

None

2.1.10. Standalone Mode

Definition:

Single controller handling all control plane functionalities without redundancy, or the ability to provide high availability and/or automatic failover.

Discussion:

In standalone mode, one controller manages one or more network domains.

Measurement Units:

N/A

See Also:

None

2.1.11. Cluster/Redundancy Mode

Definition:

A group of 2 or more controllers handling all control plane functionalities.

Discussion:

In cluster mode, multiple controllers are teamed together for the purpose of load sharing and/or high availability. The controllers in

the group may work in active/standby (master/slave) or active/active (equal) mode depending on the intended purpose.

Measurement Units:

N/A

See Also:

None

2.1.12. Asynchronous Message

Definition:

Any message from the SDN node that is generated for network events.

Discussion:

Control messages like flow setup request and response message is classified as asynchronous message. The controller has to return a response message. Note that the SDN node will not be in blocking mode and continues to send/receive other control messages

Measurement Units:

N/A

See Also:

None

2.1.13. Test Traffic Generator

Definition:

Test Traffic Generator is an entity that generates/receives network traffic.

Discussion:

Test Traffic Generator can be an entity that interfaces with SDN Nodes to send/receive real-time network traffic.

Measurement Units:

N/A

See Also:

None

2.2. Test Configuration/Setup Terms

2.2.1. Number of SDN Nodes

Definition:

The number of SDN nodes present in the defined test topology.

Discussion:

The SDN nodes defined in the test topology can be deployed using real hardware or emulated in hardware platforms.

Measurement Units:

N/A

See Also:

None

2.2.2. Test Iterations

Definition:

The number of times the test needs to be repeated.

Discussion:

The test needs to be repeated for multiple iterations to obtain a reliable metric. It is recommended that this test SHOULD be performed for at least 10 iterations to increase the confidence in measured result.

Measurement Units:

N/A

See Also:

None

2.2.3. Test Duration

Definition:

Defines the duration of test trails for each iteration.

Discussion:

Test duration forms the basis for stop criteria for benchmarking tests. Test not completed within this time interval is considered as incomplete.

Measurement Units:
seconds

See Also:
None

2.2.4. Number of Cluster nodes

Definition:

Defines the number of controllers present in the controller cluster.

Discussion:

This parameter is relevant when testing the controller performance in clustering/teaming mode. The number of nodes in the cluster MUST be greater than 1.

Measurement Units:
N/A

See Also:
None

2.3. Benchmarking Terms

This section defines metrics for benchmarking the SDN controller. The procedure to perform the defined metrics is defined in the accompanying methodology document.

2.3.1. Performance

2.3.1.1. Network Topology Discovery Time

Definition:

To measure the time taken to discover the network topology - nodes and links by a controller.

Discussion:

Network topology discovery is key for the SDN controller to provision and manage the network. So it is important to measure how quickly the controller discovers the topology to learn the current network state. This benchmark is obtained by presenting a network topology (Tree, Mesh or Linear) with the given number of nodes to the controller and wait for the discovery process to complete. It is expected that the controller supports network discovery mechanism and uses protocol messages for its discovery process.

Measurement Units:
milliseconds

See Also:
None

2.3.1.2. Asynchronous Message Processing Time

Definition:

To measure the time taken by the controller to process an asynchronous message.

Discussion:

For SDN to support dynamic network provisioning, it is important to measure how quickly the controller responds to an event triggered from the network. The event could be any notification messages generated by an SDN node upon arrival of a new flow, link down etc. This benchmark is obtained by sending asynchronous messages from every connected SDN nodes one at a time for the defined test duration. This test assumes that the controller will respond to the received asynchronous message.

Measurement Units:
milliseconds

See Also:
None

2.3.1.3. Asynchronous Message Processing Rate

Definition:

To measure the maximum number of asynchronous messages that a controller can process within the test duration.

Discussion:

As SDN assures flexible network and agile provisioning, it is important to measure how many network events that the controller can handle at a time. This benchmark is obtained by sending asynchronous messages from every connected SDN nodes at full connection capacity for the given test duration. This test assumes that the controller will respond to all the received asynchronous messages.

Measurement Units:
Messages processed per second.

See Also:

None

2.3.1.4. Reactive Path Provisioning Time

Definition:

The time taken by the controller to setup a path reactively between source and destination node, expressed in milliseconds.

Discussion:

As SDN supports agile provisioning, it is important to measure how fast that the controller provisions an end-to-end flow in the dataplane. The benchmark is obtained by sending traffic from a source endpoint to the destination endpoint, finding the time difference between the first and the last flow provisioning message exchanged between the controller and the SDN nodes for the traffic path.

Measurement Units:

milliseconds.

See Also:

None

2.3.1.5. Proactive Path Provisioning Time

Definition:

The time taken by the controller to setup a path proactively between source and destination node, expressed in milliseconds.

Discussion:

For SDN to support pre-provisioning of traffic path from application, it is important to measure how fast that the controller provisions an end-to-end flow in the dataplane. The benchmark is obtained by provisioning a flow on controller's northbound interface for the traffic to reach from a source to a destination endpoint, finding the time difference between the first and the last flow provisioning message exchanged between the controller and the SDN nodes for the traffic path.

Measurement Units:

milliseconds.

See Also:

None

2.3.1.6. Reactive Path Provisioning Rate

Definition:

Measure the maximum number of independent paths a controller can concurrently establish between source and destination nodes reactively within the test duration, expressed in paths per second.

Discussion:

For SDN to support agile traffic forwarding, it is important to measure how many end-to-end flows that the controller could setup in the dataplane. This benchmark is obtained by sending traffic each with unique source and destination pairs from the source SDN node and determine the number of frames received at the destination SDN node.

Measurement Units:

Paths provisioned per second.

See Also:

None

2.3.1.7. Proactive Path Provisioning Rate

Definition:

Measure the maximum number of independent paths a controller can concurrently establish between source and destination nodes proactively within the test duration, expressed in paths per second.

Discussion:

For SDN to support pre-provisioning of traffic path for a larger network from the application, it is important to measure how many end-to-end flows that the controller could setup in the dataplane. This benchmark is obtained by sending traffic each with unique source and destination pairs from the source SDN node. Program the flows on controller's northbound interface for traffic to reach from each of the unique source and destination pairs and determine the number of frames received at the destination SDN node.

Measurement Units:

Paths provisioned per second.

See Also:

None

2.3.1.8. Network Topology Change Detection Time

Definition:

The amount of time required for the controller to detect any changes in the network topology.

Discussion:

In order to for the controller to support fast network failure recovery, it is critical to measure how fast the controller is able to detect any network-state change events. This benchmark is obtained by triggering a topology change event and measuring the time controller takes to detect and initiate a topology re-discovery process.

Measurement Units:
milliseconds

See Also:
None

2.3.2. Scalability

2.3.2.1. Control Sessions Capacity

Definition:

To measure the maximum number of control sessions the controller can maintain.

Discussion:

Measuring the controller's control sessions capacity is important to determine the controller's system and bandwidth resource requirements. This benchmark is obtained by establishing control session with the controller from each of the SDN node until it fails. The number of sessions that were successfully established will provide the Control Sessions Capacity.

Measurement Units:
N/A

See Also:
None

2.3.2.2. Network Discovery Size

Definition:

To measure the network size (number of nodes, links and hosts) that a controller can discover.

Discussion:

For optimal network planning, it is key to measure the maximum network size that the controller can discover. This benchmark is obtained by presenting an initial set of SDN nodes for discovery to the controller. Based on the initial discovery, the number of SDN nodes is increased or decreased to determine the maximum nodes that the controller can discover.

Measurement Units:

N/A

See Also:

None

2.3.2.3. Forwarding Table Capacity

Definition:

The maximum number of flow entries that a controller can manage in its Forwarding table.

Discussion:

It is significant to measure the capacity of controller's Forwarding Table to determine the number of flows that controller could forward without flooding/dropping. This benchmark is obtained by continuously presenting the controller with new flow entries through reactive or proactive flow provisioning mode until the forwarding table becomes full. The maximum number of nodes that the controller can hold in its Forwarding Table will provide Forwarding Table Capacity.

Measurement Units:

Maximum number of flow entries managed.

See Also:

None

2.3.3. Security

2.3.3.1. Exception Handling

Definition:

To determine the effect of handling error packets and notifications on performance tests.

Discussion:

This benchmark test is to be performed after obtaining the baseline performance of the performance tests defined in Section 2.3.1. This

benchmark determines the deviation from the baseline performance due to the handling of error or failure messages from the connected SDN nodes.

Measurement Units:

N/A

See Also:

None

2.3.3.2. Denial of Service Handling

Definition:

To determine the effect of handling denial of service (DoS) attacks on performance and scalability tests.

Discussion:

This benchmark test is to be performed after obtaining the baseline performance of the performance and scalability tests defined in section 2.3.1 and section 2.3.1.. This benchmark determines the deviation from the baseline performance due to the handling of denial of service attacks on controller.

Measurement Units:

Deviation of baseline metrics while handling Denial of Service Attacks.

See Also:

None

2.3.4. Reliability

2.3.4.1. Controller Failover Time

Definition:

The time taken to switch from an active controller to the backup controller, when the controllers work in redundancy mode and the active controller fails.

Discussion:

This benchmark determine the impact of provisioning new flows when controllers are teamed and the active controller fails.

Measurement Units:

milliseconds.

See Also:

None

2.3.4.2. Network Re-Provisioning Time

Definition:

The time taken to re-route the traffic by the Controller, when there is a failure in existing traffic paths.

Discussion:

This benchmark determines the controller's re-provisioning ability upon network failures. This benchmark test assumes the following:

- i. Network topology supports redundant path between source and destination endpoints.
- ii. Controller does not pre-provision the redundant path.

Measurement Units:

milliseconds.

See Also:

None

3. Test Coverage

+-----+-----+-----+-----+-----+				
		Speed	Scalability	Reliability
+-----+-----+-----+-----+-----+				
Setup	1. Network Topology Discovery		1. Network Discovery Size	
	2. Reactive Path Provisioning Time			
	3. Proactive Path Provisioning Time			
	4. Reactive Path Provisioning Rate			
	5. Proactive Path Provisioning Rate			

Operational	<ol style="list-style-type: none"> 1. Asynchronous Message Processing Rate 2. Asynchronous Message Processing Time 	<ol style="list-style-type: none"> 1. Control Sessions Capacity 2. Forwarding Table Capacity 	<ol style="list-style-type: none"> 1. Network Topology Change Detection Time 2. Exception Handling 3. Denial of Service Handling 4. Network Re-Provisioning Time
Tear Down			<ol style="list-style-type: none"> 1. Controller Failover Time

4. References

4.1. Normative References

- [RFC2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC6241] R. Enns, M. Bjorklund, J. Schoenwaelder, A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC6020] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010
- [RFC5440] JP. Vasseur, JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [OpenFlow Switch Specification] ONF, "OpenFlow Switch Specification" Version 1.4.0 (Wire Protocol 0x05), October 14, 2013.
- [I-D.sdn-controller-benchmark-meth] Bhuvaneshwaran.V, Anton Basil,

Mark.T, Vishwas Manral, Sarah Banks "Benchmarking
Methodology for SDN Controller Performance",
draft-ietf-bmwg-sdn-controller-benchmark-meth-00
(Work in progress), October 19, 2015

4.2. Informative References

[OpenContrail] Ankur Singla, Bruno Rijsman, "OpenContrail
Architecture Documentation",
<http://opencontrail.org/opencontrail-architecture-documentation>

[OpenDaylight] OpenDaylight Controller:Architectural Framework,
https://wiki.opendaylight.org/view/OpenDaylight_Controller

5. IANA Considerations

This document does not have any IANA requests.

6. Security Considerations

Security issues are not discussed in this memo.

7. Acknowledgements

The authors would like to acknowledge Sandeep Gangadharan (HP) for the significant contributions to the earlier versions of this document. The authors would like to thank the following individuals for providing their valuable comments to the earlier versions of this document: Al Morton (AT&T), M. Georgescu (NAIST), Andrew McGregor (Google), Scott Bradner (Harvard University), Jay Karthik (Cisco), Ramakrishnan (Dell), Khasanov Boris (Huawei).

8. Authors' Addresses

Bhuvaneswaran Vengainathan
Veryx Technologies Inc.
1 International Plaza, Suite 550
Philadelphia
PA 19113

Email: bhuvaneswaran.vengainathan@veryxtech.com

Anton Basil
Veryx Technologies Inc.
1 International Plaza, Suite 550

Philadelphia
PA 19113

Email: anton.basil@veryxtech.com

Mark Tassinari
Hewlett-Packard,
8000 Foothills Blvd,
Roseville, CA 95747

Email: mark.tassinari@hp.com

Vishwas Manral
Ionos Corp,
4100 Moorpark Ave,
San Jose, CA

Email: vishwas@ionosnetworks.com

Sarah Banks
VSS Monitoring
930 De Guigne Drive,
Sunnyvale, CA

Email: sbanks@encrypted.net

